

**Title: The FrameWork: An Open-Architecture for Very Large Image
Exploitation**

Track submitted for: Experimentation

Authors: George O. Ramseyer, Scott E. Spetka, Richard E. Linderman and Brian C. Romano

Dr. George Ramseyer
Air Force Research Laboratory/Information Directorate
26 Electronics Parkway
Rome, NY 13441-4514
Phone: (315) 330-3492
George.Ramseyer@rl.af.mil

Dr. Scott E. Spetka
ITT Industries
Advanced Engineering and Sciences
SUNY Institute of Technology
PO Box 3050
Utica, NY 13504
scott@cs.sunyit.edu

Dr. Richard E. Linderman
Air Force Research Laboratory/Information Directorate
26 Electronics Parkway
Rome, NY 13441-4514
Phone: (315) 330-2208
Richard.Linderman@rl.af.mil

Brian C. Romano
Air Force Research Laboratory/Information Directorate
32 Brooks Road
Rome, NY 13441-4118
Phone: (315) 330-4218
George.Ramseyer@rl.af.mil

The FrameWork: An Open-Architecture for Very Large Image Exploitation

George O. Ramseyer, Scott E. Spetka*, Richard E. Linderman and Brian C. Romano

Air Force Research Laboratory/Information Directorate

26 Electronics Parkway

Rome, NY 13441-4514

Phone: (315) 330-3492

George.Ramseyer@rl.af.mil

* ITT Industries

Advanced Engineering and Sciences

SUNY Institute of Technology

PO Box 3050

Utica, NY 13504

scott@cs.sunyit.edu

Abstract

Advances in sensor collection and display technologies have led to increasingly larger imagery databases with increasingly complex processing requirements. Combining high-speed communications and remote parallel processing computers through shared access, sensor data can be transmitted and processed, and the resultant imagery products can then be efficiently disseminated through information-based systems. The FrameWork is a system that exploits very large image processing technology through Broadsword, and is being designed to interface with information-based Command and Control systems that are under development, such as the Joint Battlespace Infosphere.

Introduction

Effective Command and Control begins with battlespace monitoring. Sensor data collected from relevant aspects of the battlefield can be transformed into databases of imagery products, and tools for data collection and processing for all types of images are under development. Many of these imagery systems are unique, and the integration of those systems with other systems has in many cases been challenging.

Photographic imagery can be used to identify and extract characterizing attributes. Densities of trees or availability of water can be gleaned from photographs. Visual analysis can fall short in identifying some of the characteristics in a photograph. For example, trees and leaves may be

Report Documentation Page			Form Approved OMB No. 0704-0188					
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>								
1. REPORT DATE 2002	2. REPORT TYPE	3. DATES COVERED 00-00-2002 to 00-00-2002						
4. TITLE AND SUBTITLE The FrameWork: An Open-Architecture for Very Large Image Exploitation			5a. CONTRACT NUMBER					
			5b. GRANT NUMBER					
			5c. PROGRAM ELEMENT NUMBER					
6. AUTHOR(S)			5d. PROJECT NUMBER					
			5e. TASK NUMBER					
			5f. WORK UNIT NUMBER					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFTC,26 Electronic Parkway,Rome,NY,13441			8. PERFORMING ORGANIZATION REPORT NUMBER					
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)					
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)					
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited								
13. SUPPLEMENTARY NOTES 2002 Command & Control Research & Technology Symposium								
14. ABSTRACT								
15. SUBJECT TERMS								
16. SECURITY CLASSIFICATION OF: <table border="1"> <tr> <td>a. REPORT unclassified</td> <td>b. ABSTRACT unclassified</td> <td>c. THIS PAGE unclassified</td> </tr> </table>			a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 7	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified						

identified, but the type of tree may not be visually identifiable. New technologies allow additional information content to be gleaned from images through automated processes. Image processing algorithms help extend human recognition through the application of pattern recognition.

Although algorithms have been developed that can process very large images, a system is needed to exploit these technologies. The system must be capable of performing the required imagery analysis and must also be defined with sufficient network capacity to deliver imagery to points of service and provide for collaboration. Due to the intense processing requirements related to very large image algorithms, high-performance parallel computing systems are essential.

The FrameWork Architecture is designed to support algorithms hosted on tightly coupled architectures, such as Sky and Mercury high performance computing systems and modern cluster Linux Beowulf computing systems. In addition to intense processing requirements, high-performance network support will also be needed to transfer very large images to high-performance computing centers where algorithms may be supported.

Sharing Software and HPCs - Open Architecture

Very large image processing algorithms and high-performance computers are integrated into a flexible architecture that provides an opportunity for expansion while offering the performance and flexibility required to meet the needs of Command and Control. A range of algorithms has been developed to serve different purposes, and is available to those with access and who are familiar with their functionality. The architecture allows new algorithms to be installed without disrupting access to existing algorithms. The Architecture depicted in Figure 1 supports sharing software across available high performance computers through a high-speed network.

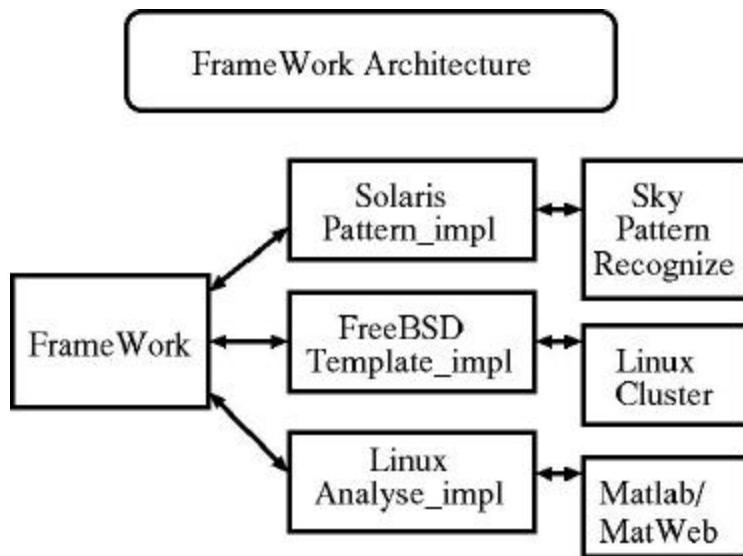


Figure 1 - The FrameWork Architecture

Figure 1 shows very large image processing algorithms executing on HPCs and accessed through host computers using the Solaris and Linux operating systems. Raw images can be processed for

pattern recognition through software executing on a Sky computer with the implementation hosted on a Solaris system. The resulting image can be processed on a Linux system hosting a Beowulf cluster with MATLAB (1) installed on the cluster computers. The MATLAB implementation allows querying the characteristics of an image that may be significant, but which are not readily visibly discernible. A “Template” object implementation is used to design an algorithm’s interface to the FrameWork client.

Image characteristics recognized as important, based on very large image analysis, must be shared with others to be useful in a decision-making process. This requires the translation of imagery products into reports that can be readily understood by those in Command and Control. Very large image data semantics often cannot be inferred from a visual examination of an image display, and systematic analysis by pattern recognition algorithms is necessary.

The FrameWork is designed to interface to systems for sharing images. The Broadsword (2) interface allows images to be cataloged into a variety of heterogeneous database systems, including the National Imagery and Mapping Agency’s (NIMA’s) Image Product Library (IPL). Currently under development is the FrameWork interface to the Joint Battlespace Infosphere (JBI) (3). When completed, this FrameWork/JBI system will allow raw images and processed images to be published for possible downloading by interested parties, including those from the Command and Control community. Figure 2 shows how very large imagery data would be exploited with the FrameWork.

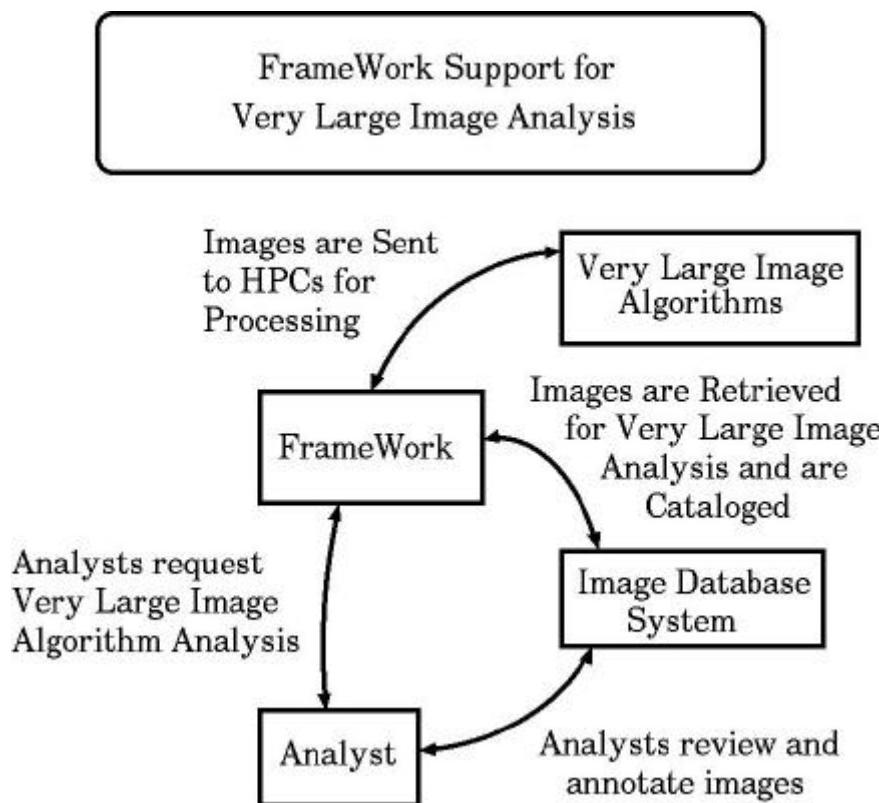


Figure 2 - Using the FrameWork to Share Very Large Images

Imagery data is requested utilizing Broadsword or in the future an information system such as the Joint Battlespace Infosphere. After the imagery data is processed to form an imagery product, the product may be annotated, noting for instance features that are recognized. Then the imagery product may be catalogued into large databases using Broadsword or, when development is complete, re-published through the JBI. If the user is authorized to query other databases, additional imagery products or reports may be requested.

FrameWork Implementation

The FrameWork, an overview of which is presented in Figure 3, is built using technologies that allow for flexibility and reliability. The system is driven by CGI-BIN Web interface software (4) that is installed along with a web server. The Web-based implementation allows access to any authorized user who has a Web browser and an Internet connection. The implementation of new algorithms can be incorporated into the system without disrupting normal system operation. Common Object Request Broker Architecture (CORBA) software (5) is used to allow new algorithms to be added to the system. Each image processing object implementation is accessed by the FrameWork client via the Common Object Services (COS) Naming Server (6). Servers that implement very large image processing algorithms register themselves with the COS Naming Server when they become available.

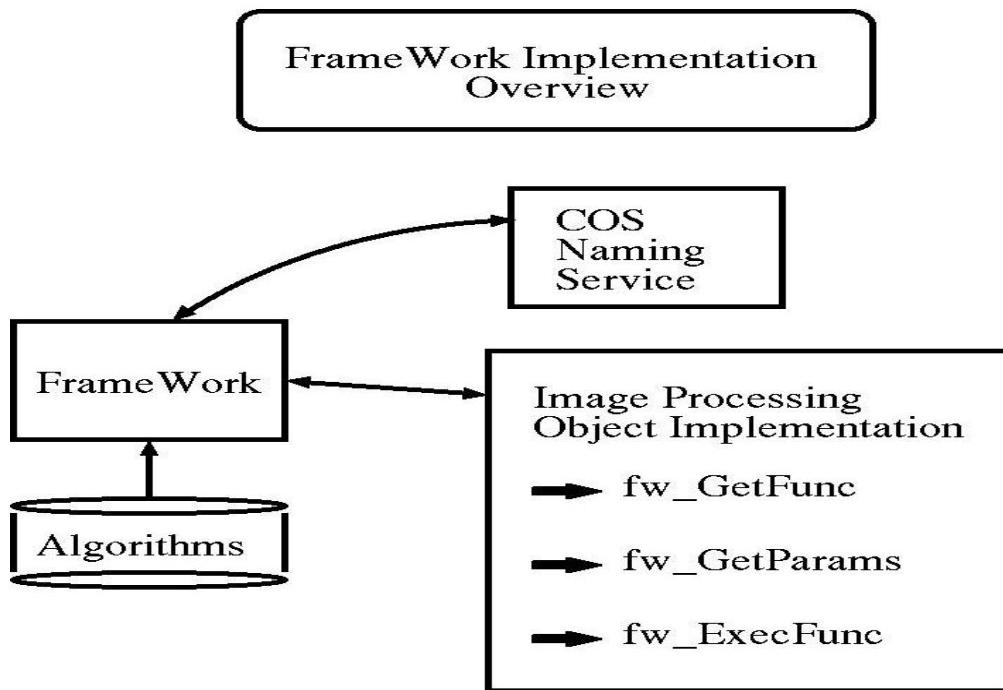


Figure 3 - FrameWork Implementation Overview

The FrameWork uses the COS naming service to connect to the algorithm server implementation. An administrative function is performed for the Framework to configure the names of available algorithms. The names are displayed for the user to select through the Framework client. There are

three main functions implemented by each algorithm that is accessible through the FrameWork client.

Figure 4 presents the FrameWork implementation in further detail. Initially, when an object which implements an algorithm becomes available to the FrameWork client, it registers itself with the COS naming service. If the FrameWork client has that algorithm configured for access, it can retrieve an object reference that can be used to invoke the algorithm from the COS naming service. In the initial implementation of the FrameWork, the available algorithms are configured into an object that is capable of displaying them for the Framework client.

The client uses the object reference to retrieve a list of functions from the algorithm object that allows the user to select one of the functions implemented by the algorithm. The Framework then sends a request to the algorithm implementation to determine the parameters that are needed by the function. The response from the object must be formatted to describe the input structures required by the algorithm. Inputs can be files, selected from specified directories (or folders) or input by the user along with a directory location. A list of files in a selected output directory is displayed for the convenience of the users. An algorithm may also request inputs via select boxes, check boxes and radio buttons when limited inputs are allowed. Text inputs where users may type any input they wish are also available to the algorithm designer.

FrameWork Implementation Detail		
FrameWork	Algorithm Implementation	
FrameWork Client Retrieves List of Available Algorithms	Corba Object Implementation Registers with COS Naming	
FW Displays Available Algorithms	User Selects Algorithm	fw_GetFunc: Returns a list of available functions
FW Displays Available Functions	User Selects Function	fw_GetParams: Returns Tagged Parameters for Required Inputs (Control of HTML Display)
FW Displays Input Request Form	User Inputs Parameters for Selected Function	fw_ExecFunc: Executes Function and Sends an HTML Response
FW Displays Algorithm Output		

Figure 4 - FrameWork Implementation Details

The parameters are passed to the algorithm for execution after all inputs are specified. The object executes the function and constructs a response string that is displayed by the FrameWork client. The response string may contain the HTML necessary to display a button or hypertext link that can

be used to view a resulting image. Output tables are another of the unlimited options available through the FrameWork's open architecture.

Conclusion

This paper describes an open systems architecture that provides a foundation for very large image exploitation. High-speed communications allows the processing of imagery data on remote parallel processing computers. The resultant imagery products are then efficiently disseminated to decision makers. Thus, with an open systems architecture the FrameWork enables processing resources for very large image exploitation to be fully utilized in information-based Command and Control systems.

References

- 1) <http://www.mathworks.com/> accessed 4 April 2002.
- 2) <http://extranet.if.afrl.af.mil/bsword/> accessed 1 April 2002.
- 3) <http://spock.deepthought.rl.af.mil/programs/jbi/> /accessed 1 April 2002.
- 4) <http://www.execpc.com/~keithp/bdlogcgi.htm> / accessed 4 April 2002.
- 5) <http://www.omg.org/> / accessed 1 April 2002.
- 6) <http://java.sun.com/products/jdk/1.2/docs/guide/idl/jidlNaming.html> /accessed 1 April 2002.

Acknowledgement

This work was sponsored in part by the Department of Defense High Performance Computing Modernization Program's Common High Performance Computing Software Support Initiative (CHSSI).